ÍNDICE

ÍNDICE

ÍNDICE 1

A. EL *SOFTWARE* DE PROGRAMACIÓN CODESYS 3

A.1. Introducción a *CoDeSys* V2.3 3
A.2. Requisitos mínimos del sistema 4
A.3. Descripción general del *Software* 4
A.4. Primeros pasos 6
A.5. Verificación del programa 9
A.6. Configuración del autómata 10
A.7. Configuración de las comunicaciones de red 12
A.8. Descarga del programa al autómata 14
A.9. Menú ayuda 15

B. PRÁCTICAS DE LA PLATAFORMA 17

B.1. Práctica 1 18
B.2. Práctica 2 22
B.3. Practica 3 25
B.4. Práctica 4 26

C. VALORACIÓN ECONÓMICA 31

C.1. Coste del prototipo 31
C.2. Coste de investigación y desarrollo 32
C.3. Coste total del proyecto 33

D. ESTUDIO DEL IMPACTO AMBIENTAL 35

D.1. Materiales utilizados 35
D.2. Reciclaje 36

BIBLIOGRAFÍA 37

Referencias bibliográficas 37

A. El *software* de programación CoDeSyS

En este capítulo se dan las directrices básicas para la utilización del *software* CoDeSyS V2.3. Este programa permite crear los programas de control de las estaciones didácticas de la plataforma, además de comunicarse con el autómata, vía *Ethernet*, y transferirle la información [1].

A.1. Introducción a *CoDeSys* V2.3

CoDeSys es uno de los sistemas de programación del estándar IEC 61131-3 más comunes para PLCs y reguladores industriales. Permite programar autómatas de diferentes fabricantes, así cómo generar código para algunos microprocesadores y DSPs.

Fig. A.1 Logo e información del programa CoDeSys.

Soporta los cinco lenguajes de programación del estándar, dispone de visualización integrada, además de un simulador *offline*.

Se trata de un *software* de programación abierto y se puede descargar, incluyendo un manual de ayuda en línea muy completo, directamente de la página del fabricante de los PLCs, www.wago.com.

A.2. Requisitos mínimos del sistema

Para poder utilizar este *software* sin problemas se requiere tener un sistema con las siguientes características como mínimo:

- *Pentium* II, 500 MHz
- 128 MB RAM
- 70 MB de espacio de disco duro disponible
- *Windows* 98 / NT 4.0 / 2000 / XP (*MS Internet Explorer* 4.0 o posterior)
- CD ROM *drive*

A.3. Descripción general del *Software*

CoDeSyS V2.3 es el *software* destinado a la creación de los programas del autómata para controlar las diferentes estaciones de la plataforma.

Existen diferentes menús de trabajo –Fig. A.2– y a continuación se hace una pequeña explicación de los mismos:

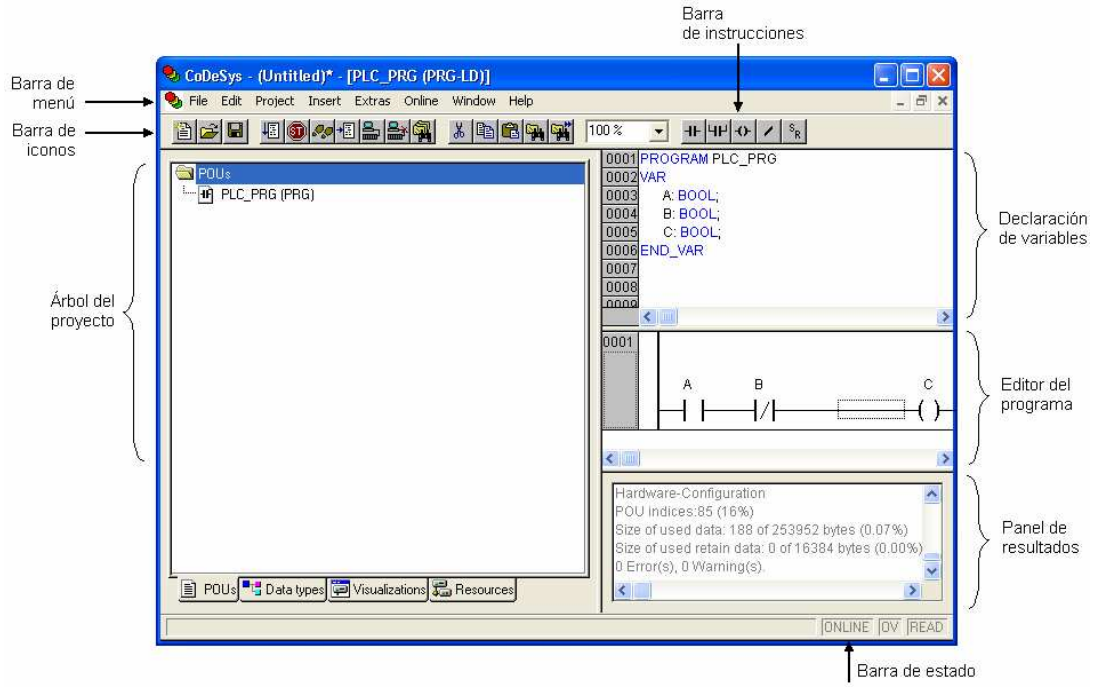

Fig. A.2 Ventana general del programa CoDeSys.

Barra de menú: permite realizar diferentes funciones como recuperar o guardar programas, opciones de ayuda, etc. Es decir, las funciones elementales de cualquier *software* actual.

Barra de iconos: engloba las funciones de uso más repetido en el desarrollo de los programas.

Barra de instrucciones: Esta barra le permitirá, a través de pestañas y botones, acceder de forma rápida a las instrucciones más habituales del lenguaje utilizado (en el ejemplo de la figura A2 –Fig. A.2– se utiliza lenguaje *Ladder*). Presionando sobre cada instrucción, ésta se introducirá en el editor del programa.

Panel de resultados: aparecen los errores de programación que surgen al verificar la corrección del programa realizado (*Project* > *Build*) o del proyecto completo (*Project* > *Rebuild all*). Efectuando doble clic sobre el error, automáticamente el cursor se sitúa sobre la ventana del editor del programa en la posición donde se ha producido el error.

Declaración de variables: En esta ventana aparecen todas las variables definidas en el programa, así como la declaración de todos los temporizadores, contadores u otros objetos que se hayan utilizado. También es posible la inicialización de una variable, así como la asignación de una salida o una entrada, digital o analógica del autómata.

Editor del programa: Contiene el texto o el esquema, según el lenguaje utilizado entre los 5 lenguajes que soporta *CoDeSyS*, del programa que se esté realizando. Se puede interaccionar sobre esta ventana escribiendo el programa directamente desde el teclado o ayudándose con el ratón (ya sea arrastrando objetos procedentes de otras ventanas ó seleccionando opciones con el botón derecho del ratón).

Barra de estado: Nos permite visualizar el modo de trabajo del procesador (*online*/*offline*). En el caso de estar online, la palabra *ONLINE* se pone en negrita y se puede visualizar a continuación cuál es el autómata con el que estamos comunicados o si estamos en modo simulación, activándose la palabra *SIM*. Si el autómata o la simulación están en marcha, aparecerá la palabra *RUNNING* en negrita.

Árbol del proyecto: Esta ventana está situada en la izquierda de la pantalla –Fig. A.2–. En el fondo dispone de cuatro pestañas cada una de las cuales tiene una función diferente, son las siguientes:

- POUs (*Program Organization Unit*): Tanto funciones como bloques de funciones y programas son POUs, los cuales pueden ser complementados por acciones. Cada POU consiste en una parte de declaración y un cuerpo. El cuerpo del programa se escribe en el editor del programa en uno de los lenguajes de programación del IEC, que son IL, ST, SFC, FBD, LD o CFC.

  CoDeSys soporta todos los POUs del estándar IES. Si se quieren usar estos POUs en el proyecto, se debe incluir la librería *standard.lib* en éste.

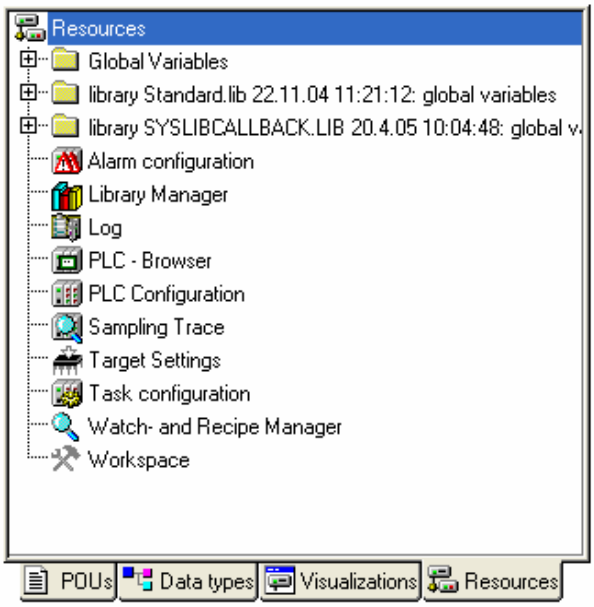

Fig. A.3 Árbol del proyecto de CoDeSys.

- *Data types*: en este apartado el usuario puede definir sus propios tipos de datos. Como por ejemplo: estructuras, tipos de enumeración y referencias.

- *Visualizations*: CoDeSys proporciona visualizaciones de modo que el usuario pueda mostrar las variables de su proyecto. Se pueden dibujar elementos geométricos *offline*, que en modo *online*, pueden cambiar su forma/color/texto en respuesta a valores especificados de las variables.

- *Resources*: en este apartado se ayuda a configurar y a organizar el proyecto. Es posible entre otras tareas la definición de variables globales las cuales se pueden utilizar en todo el proyecto; la gestión de las librerías; la configuración de un sistema de alarma o la configuración del hardware del PLC.

A.4. Primeros pasos

Para empezar a utilizar el programa lo primero que se debe hacer es crear un proyecto nuevo, *File* > *New* y seguidamente aparecerá un cuadro de diálogo –Fig. A.4– para configurar el autómata al que nos conectaremos. Se selecciona del menú desplegable el

siguiente: *WAGO_750-841*. Y se acepta (OK).

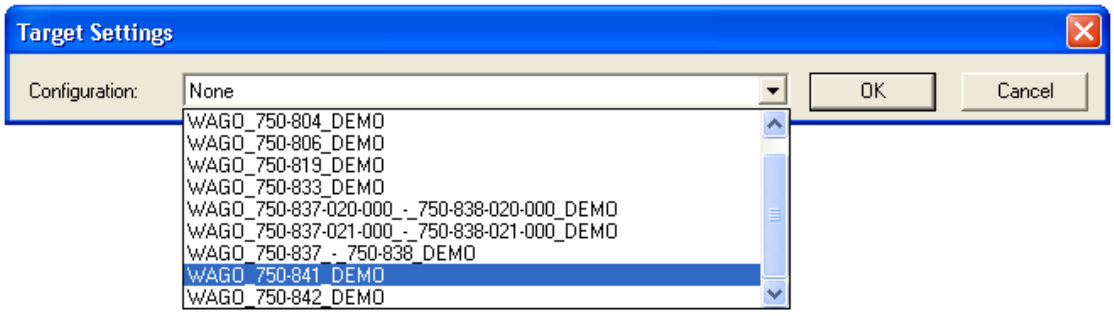

Fig. A.4 Menú para seleccionar el autómata.

A continuación aparecerá un menú para configurar el primer POU (*Program Organization Unit*) del proyecto –Fig. A.5–. Se puede elegir el lenguaje de programación entre los 5 disponibles del estándar IEC además del tipo de POU (Programa, Función o Bloque de Función).

El primer POU creado en un proyecto nuevo se llamará automáticamente PLC_PRG y este nombre no se podrá modificar.

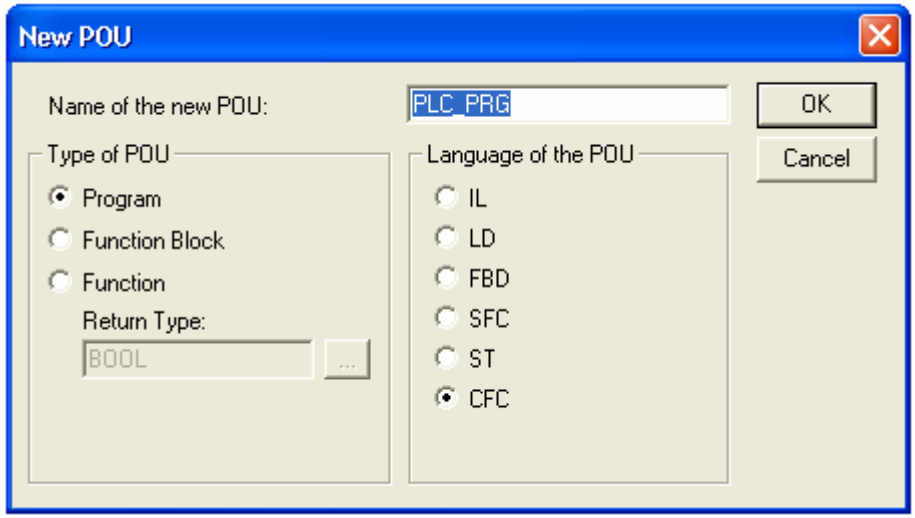

Fig. A.5 Menú de configuración del POU PLC_PRG.

Una vez creado el primer POU, ya se puede empezar a programar en el "Editor del programa" en el lenguaje deseado. La Fig. A.6 muestra un ejemplo:

```
0001 PROGRAM ORDRE
0002 VAR
0003     DECT1 AT %IX2.0: BOOL;
0004     DECT2 AT %IX2.2: BOOL;
0005     DECT3 AT %IX2.4: BOOL;
0006     SENTIT: BOOL;
0007     NUM: INT;
0008 END_VAR
0000
```

```
0001 SENTIT := FALSE;
0002 IF DECT1=TRUE THEN
0003 SENTIT := NOT(SENTIT);
0004 NUM := 0;
0005 WHILE NUM <=1000 DO
0006 NUM := NUM+1;
0007 END_WHILE;
0008 END_IF;
0009
```

Fig. A.6 Ejemplo de programación en lenguaje ST.

Para añadir más POUs al proyecto se debe acceder a la primera pestaña, POUs, del "Árbol del proyecto", hacer clic con el botón derecho del ratón en la carpeta POUs y en *Add Object*… –Fig. A.7–.

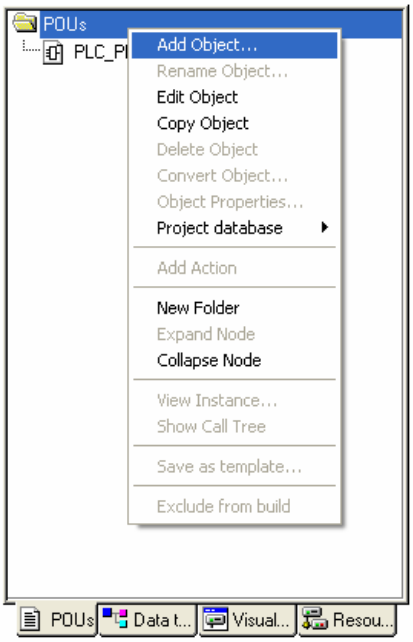

Fig. A.7 Menú para seleccionar el autómata.

A continuación aparecerá un menú idéntico al del primer POU –Fig. A.5–, para definir el tipo y el lenguaje de éste. En este caso no aparecerá ningún nombre y se podrá elegir el que se quiera (sin espacios entre medio).

Si se desea cambiar el nombre, el lenguaje, editar el objeto u otras propiedades de los POUs definidos anteriormente, se debe hacer clic con el botón derecho en el POU deseado y aparecerá un menú con todas las opciones posibles –Fig. A.8–.

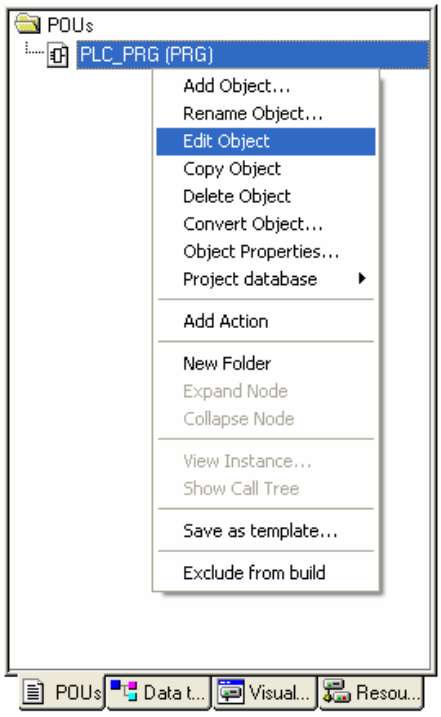

Fig. A.8 Menu para modificar las propiedades del POU.

A.5. Verificación del programa

Una vez se he ha acabado de programar el proyecto completo se debe comprobar que no contenga errores, ya sean de sintaxis como de uso incorrecto de variables.

Para verificar un proyecto se debe ir al menú *Project* > *Rebuild all*. Los posibles errores de programación aparecerán en la ventana inferior "Panel de resultados" –Fig. A.9– y mediante un doble clic en el mensaje del error se accederá directamente a la línea de programación donde se ha producido en el "Editor del programa".

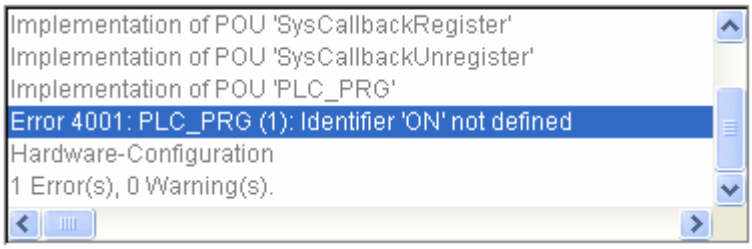

Fig. A.9 Panel de resultados donde aparecen los errores del proyecto

A.6. Configuración del autómata

Uno de los pasos más importantes a la hora de programar un PLC es su correcta configuración de *Hardware* dentro del *Software* de programación. Si este autómata dispone de conexión *Ethernet* –como es este caso– es preciso además configurar la comunicación de red.

Cada vez que se crea un proyecto nuevo, se elige la configuración adecuada del autómata (*WAGO_750-841*). Pero además es necesario configurar las tarjetas de entradas y salidas que disponemos.

Accediendo a la cuarta pestaña, *Resources*, del "Árbol del proyecto" se encuentra la opción *PLC Configuration*. Si se hace doble clic en este apartado, aparece a la derecha la configuración del *hardware* que disponemos –Fig. A.10–.

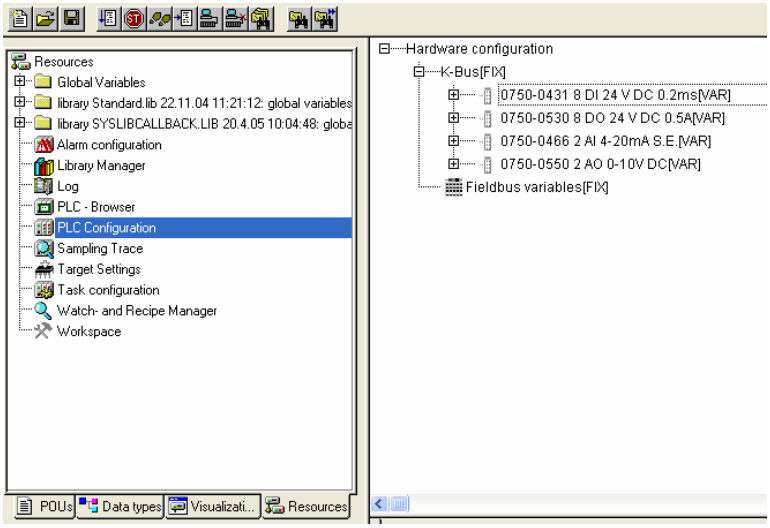

Fig. A.10 Configuración del autómata, apartado Resources.

Si en el apartado *K-Bus[FIX]* no aparecen los módulos de entradas y salidas que dispone el PLC, se deben insertar, haciendo clic con el botón derecho en *K-Bus[FIX]* y seleccionar *Insert* Element –Fig. A.11–.

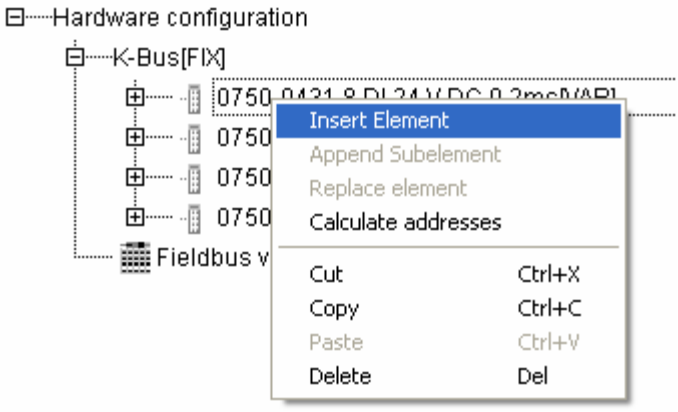

Fig. A.11 Menú para insertar los módulos de I/O del autómata.

Seguidamente aparecerá un menú de configuración de I/O donde se podrán elegir todos los módulos de entradas y salidas disponibles para el módulo ENI del autómata –Fig. A.12–.

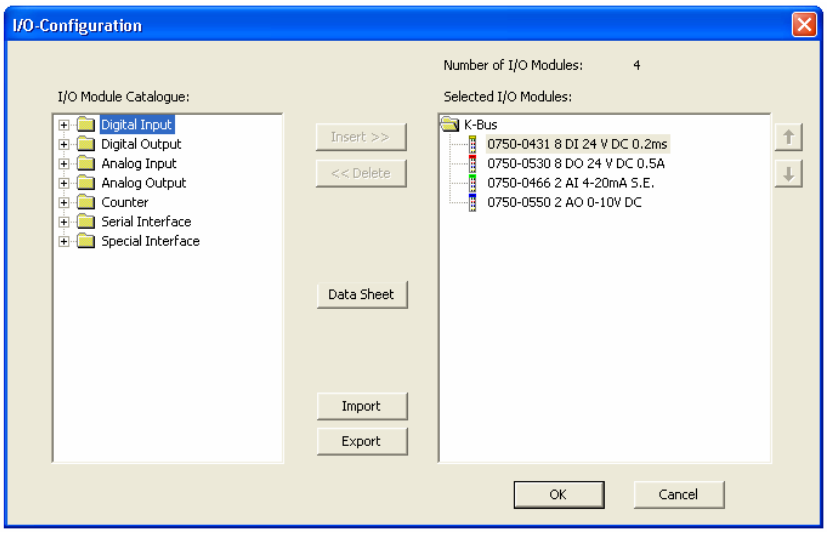

Fig. A.12 Menú de configuración de los módulos I/O del autómata.

A.7. Configuración de las comunicaciones de red

Para realizar la configuración de la red se debe acceder al apartado *Online* > *Communication Parameters* de la barra de menú –Fig. A.13–.

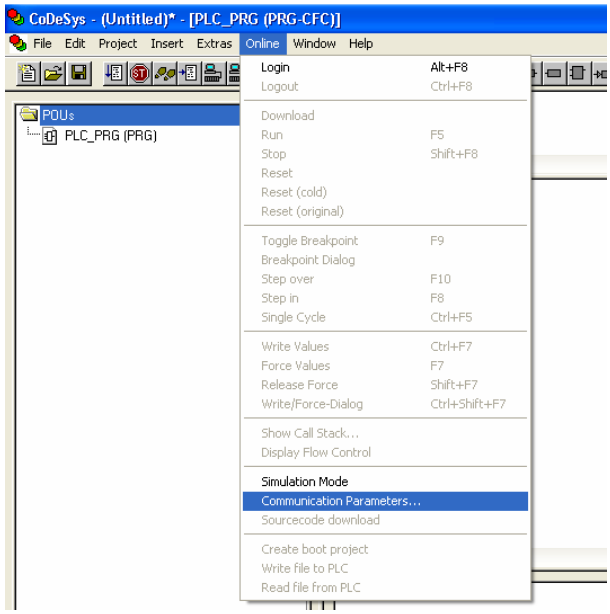

Fig. A.13 Configuración de las comunicaciones de red.

A continuación aparecerá el siguiente cuadro de diálogo –Fig. A.14–:

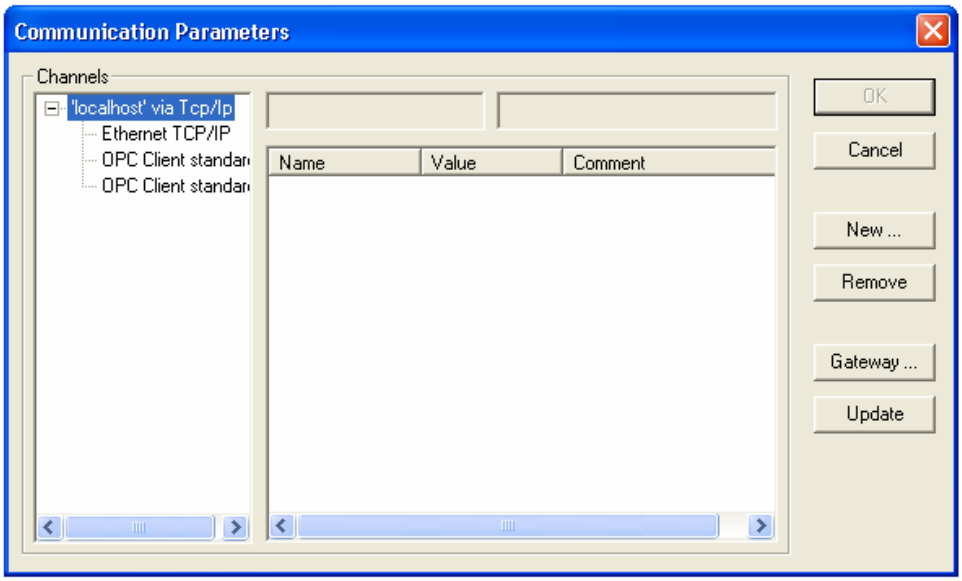

Fig. A.14 Selección de los parámetros de comunicación.

Seleccionando la opción *New*, se podrá configurar un nuevo canal de comunicaciones –Fig. A.15–. Como en la plataforma de prácticas se dispone de un autómata, se debe agregar un canal de comunicación nuevo y asignare un nombre, como por ejemplo PLC1.

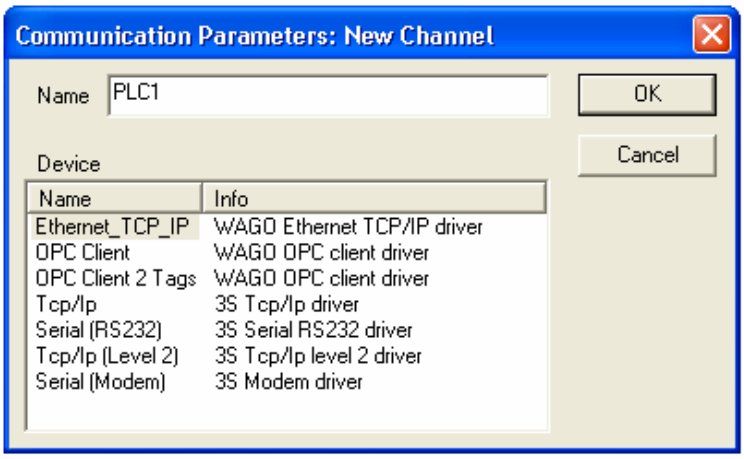

Fig. A.15 Nombre del nuevo canal de comunicación.

Al validar el nuevo canal aparecerá la configuración de la siguiente manera –Fig. A.16–:

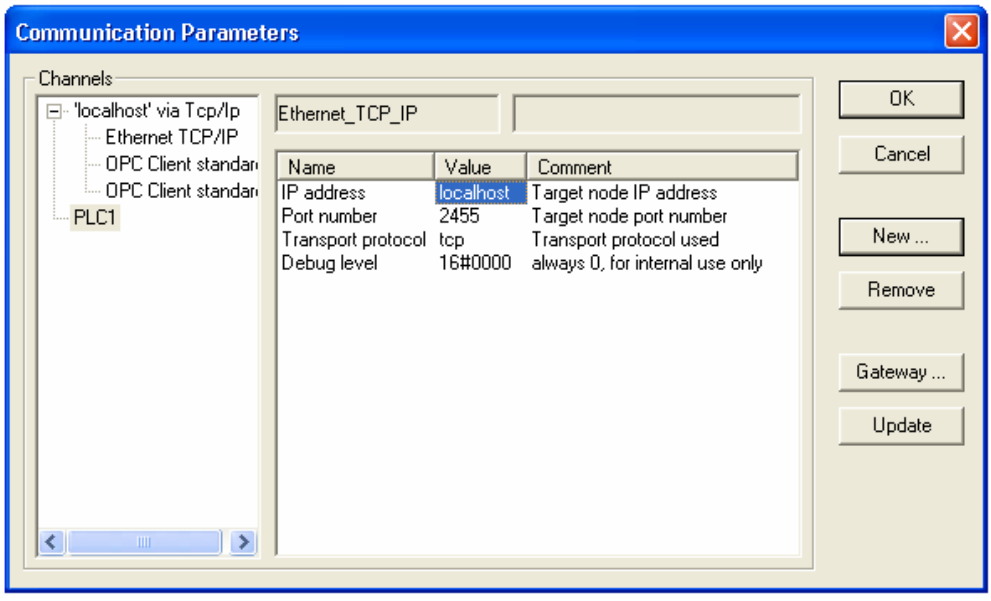

Fig. A.16 Características del canal de comunicación PLC1.

El valor de la dirección IP de la configuración PLC1 aparece vacío. Haciendo doble clic en *localhost* podemos introducir el número IP correspondiente al autómata

A.8. Descarga del programa al autómata

Una vez se ha realizado el programa y se ha verificado que no exista ningún error se procede a descargar el programa al procesador del autómata.

Se debe comprobar que el programa no este en modo de simulación, es decir que la opción *Online* > *Simulation Mode* de la barra de menú tiene que estar desactivada.

En el apartado *Online* > *Communication Parameters,* debe estar seleccionada la comunicación de red, con su correspondiente dirección IP, del PLC con el que nos queremos comunicar.

Una vez comprobadas las condiciones anteriores es posible la descarga del programa. Se debe seleccionar la opción *Online* > *Login* (o pulsando directamente *Alt+F8* o haciendo clic en el icono de la barra de iconos) y a continuación aparecerá un mensaje de confirmación de la descarga –Fig. A.17–. Seleccionamos *Yes*, y el programa se transferirá al autómata.

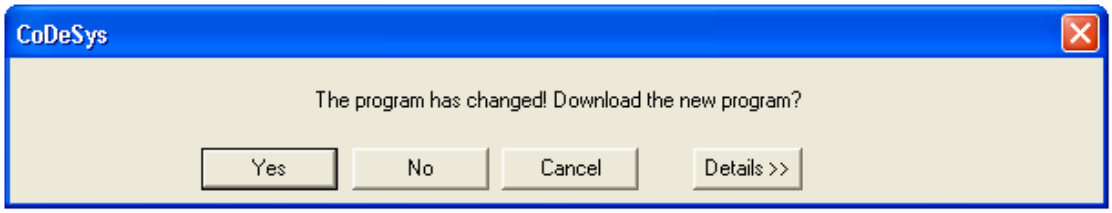

Fig. A.17 Mensaje de confirmación de la transferencia del programa al autómata.

Una vez hecha la transferencia, para poner en marcha el autómata, se debe seleccionar *Online* > *Run,* pulsar la tecla F5 o hacer clic en el icono .

De esta forma el PLC estará en modo online y las variables de entrada y salida podrán ser leídas o modificadas respectivamente. Mientras se esté conectado con el autómata se podrá observar como se va modificando el valor de las variables según la evolución del programa realizado.

Para desconectar el enlace entre el ordenador personal y el autómata se deben seguir los siguientes pasos, siempre teniendo en cuenta que una vez desconectado el autómata este sigue funcionando con el programa descargado.

Por este motivo, cada vez que sea necesario desconectar se debe seleccionar *Online* > *Stop,* pulsar *Shift*+F8 o hacer clic en el icono , para parar todos los elementos accionados durante el funcionamiento del programa.

A continuación, para interrumpir la conexión entre el autómata y el PC se debe seleccionar la opción *Online* > *Logout,* o pulsar *Ctrl+F8* o hacer clic en el icono).

A.9. Menú ayuda

Para cualquier duda que se pueda presentar en el uso del programa, se puede utilizar la ayuda que es bastante completa –apartado *Help* de la barra de menú–. Esta permite buscar según palabras clave o por agrupaciones de contenido –Fig. A.18–.

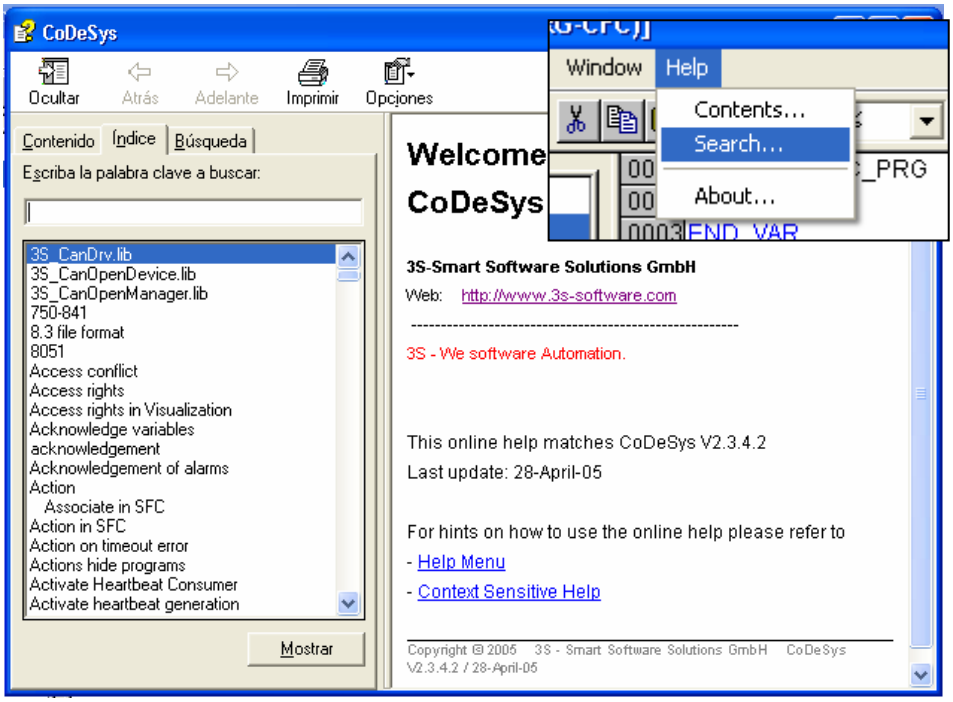

Fig. A.18 Menú de ayuda del programa.

B. Prácticas de la plataforma

En el siguiente apartado se proponen una serie de enunciados destinados a la práctica con la programación de PLC's utilizando la plataforma construida [2].

A la hora de definir estos enunciados se debe tener presente la configuración final de la plataforma y las limitaciones que esto impone. Se debe destacar que la plataforma construida basa su funcionamiento en el movimiento (vertical) de un objeto metálico a través de un campo magnético. Esta plataforma consta de dos sensores de posición digitales, los cuales proporcionan la información de la posición del objeto en dos posiciones. Se debe tener presente una diferencia entre los dos sensores, ya que el sensor situado en la parte superior (S1) tiene el emisor y el receptor por separado, mientras que el sensor de la parte inferior (S2) tiene el emisor y el receptor conjunto. El esquema es el siguiente –Fig. B.1–:

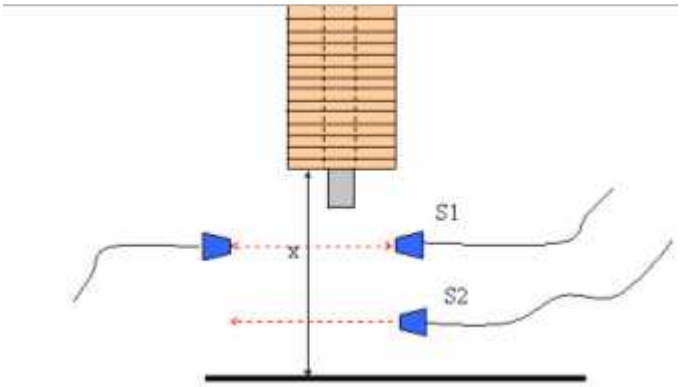

Fig. B.1 Esquema del actuador

De modo que el sensor S1 estará en estado ON en el momento en que no detecte el objeto. Sin embargo, el sensor S2 estará en estado ON en el momento en que detecte el objeto [3].

Teniendo presente la configuración de la plataforma, se llega a la conclusión de que se podrá posicionar el objeto en tres posiciones distintas:

Posición	S1	S2
Arriba	ON	OFF
En medio	OFF	OFF
Abajo	OFF	ON

Tabla B.1. Definición de las tres posiciones

B.1. Práctica 1

Como primera práctica, se propone al alumno posicionar el objeto en cada una de las tres posiciones durante un cierto periodo de tiempo ($t = 10\ s$) para después regresar a la posición inicial.

Para realizar esta práctica, se realiza un programa en lenguaje ESTRUCTURADO (ST). En este programa se definen las posibles posiciones del objeto además de especificar como se comporta la salida analógica del autómata en función de la posición del émbolo.

Se realiza una secuencia en lenguaje GRAFCET (SFC) [4] la cual define la secuencia a realizar el objeto.

Posición arriba.

En primer lugar, se define la secuencia a realizar –Fig. B.2–. Se declaran y definen las variables necesarias para confeccionar esta secuencia. Se definen dos variables tipo "tiempo" [2]. Una de ellas "tiempo inicial" que fijará el tiempo en que el émbolo permanecerá en el estado inicial, y otra "tiempo" que define el tiempo que permanece en la posición especificada. –Fig. B.3–:

```
0001 PROGRAM practicainicial1
0002 VAR
0003     tiempoinicial: TIME:=t#100ms
0004     ESTA_ARRIBA: BOOL;
0005     tiempo: TIME:=t#10s
0006 END_VAR
```

Fig. B.2 Programa Prácticainicial1_PRG. Definición de variables

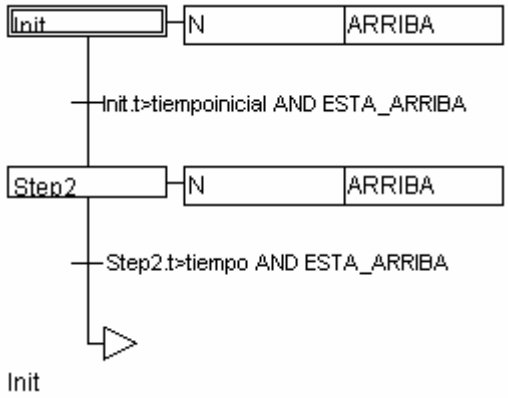

Fig. B.3 Programa Prácticainicial1_PRG. Secuencia a realizar

Observar como el estado inicial corresponde a la posición "ARRIBA"

Seguidamente, se realiza el programa en lenguaje ESTRUCTURADO (ST). En primer lugar se definen las variables necesarias –Fig. B.4–

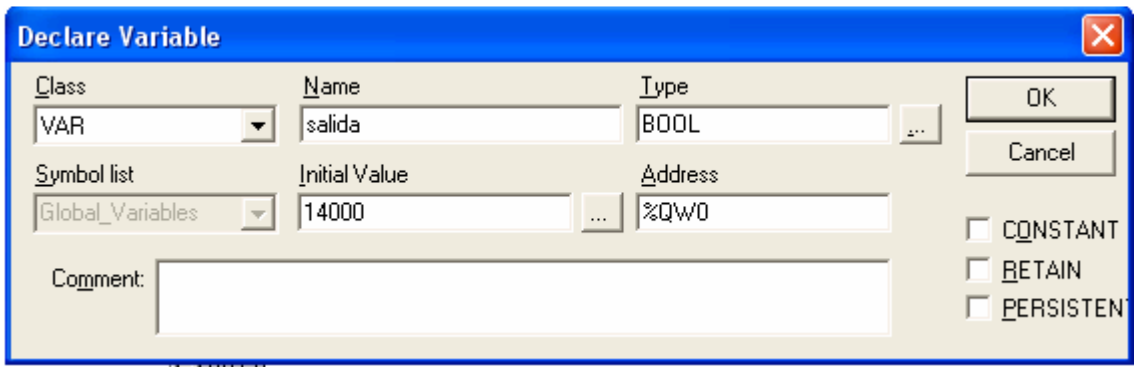

Fig. B.4 Programa PLC_PRG. Definición de variables

```
0001 PROGRAM PLC_PRG
0002 VAR
0003     ESTA_ARRIBA: BOOL;
0004     salida AT %QW0: BOOL := 14000;
0005 END_VAR
```

Fig. B.5 Programa PLC_PRG. Definición de variables

Posteriormente se define el estado. Este estado queda definido por los dos sensores de posición de que consta la plataforma –Fig. B.6–

```
0001 ESTA_ARRIBA:=%IX2.0 AND NOT (%IX2.1)
```

Fig. B.6 Programa PLC_PRG. Definición del estado

Finalmente se describe el comportamiento de la salida analógica del autómata –Fig. B.7–

```
0004 IF NOT (ESTA_ARRIBA) THEN:
0005 salida:=salida + 5
0006 END_IF
```

Fig. B.7 Programa PLC_PRG. Comportamiento de la salida

Posición abajo:

Se actúa como en el caso anterior, en primer lugar se define la secuencia a realizar (definiendo las variables adecuadas) –Fig. B.8–

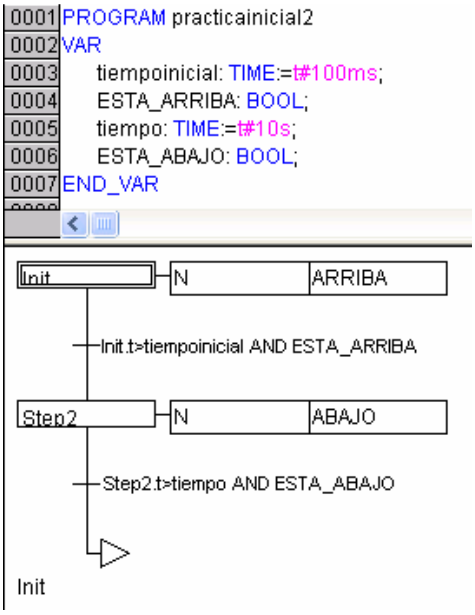

Fig. B.8 Programa practicainicial2_PRG. Secuencia a realizar

Seguidamente se implementa el programa en lenguaje ESTRUCTURADO (ST). Se declaran las variables, se definen los estados y finalmente se implementa el comportamiento de la salida analógica –Fig. B.9–:

```
0001 PROGRAM PLC_PRG
0002 VAR
0003     ESTA_ARRIBA: BOOL;
0004     ESTA_ABAJO: BOOL;
0005     salida AT %QW0: INT:=14000
0006 END_VAR
0007
```

```
0001 ESTA_ARRIBA:=%IX2.0 AND NOT (%IX2.1)
0002 ESTA_ABAJO:=%IX2.1 AND NOT (%IX2.0)
0003
0004 IF NOT (ESTA_ABAJO) THEN:
0005 salida:=salida - 5
0006 END_IF
```

Fig. B.9 Programa PLC_PRG.

Notar que en este caso es necesario definir los dos estados donde se posicionará el objeto "ARRIBA" y "ABAJO"

Posición en medio:

En este caso se procede como en los casos anteriores, si bien como se verá a continuación será necesario realizar rutinas algo más complejas.

En primer lugar se define la rutina a ejecutar (declarando las variables necesarias) –Fig. B.10–:

```
0001 PROGRAM practicainicial3
0002 VAR
0003     tiempo: TIME:=t#10s;
0004     ESTA_ENNMEDIO: BOOL;
0005     tiempoinicial: TIME:=t#100ms;
0006     ESTA_ARRIBA: BOOL;
0007 END_VAR
```

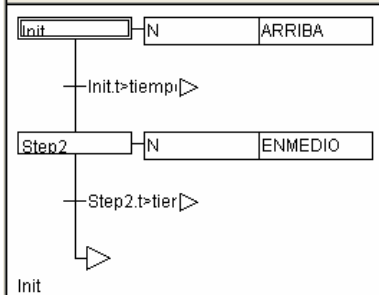

Fig. B.10 Programa PLC_PRG.

Seguidamente se implementa el programa en lenguaje ESTRUCTURADO (ST). Se declaran las variables, se definen los estados y finalmente se implementa el comportamiento de la salida analógica: –Fig. B.11–. Cabe destacar que en este caso, es necesario definir los 3 estados posibles donde se puede situar el objeto.

```
0001 PROGRAM PLC_PRG
0002 VAR
0003     ESTA_ARRIBA: BOOL;
0004     ESTA_ENMEDIO: BOOL;
0005     ESTA_ABAJO: BOOL;
0006     salida AT %QW0: INT:=14000;
0007 END_VAR
```

```
0001 ESTA_ARRIBA:=%IX2.0 AND NOT (%IX2.1)
0002 ESTA_ENMEDIO:=NOT(%IX2.0) AND NOT (%IX2.1)
0003 ESTA_ABAJO:=%IX2.1 AND NOT (%IX2.0)
0004
0005 IF NOT (ESTA_ENMEDIO) AND ESTA_ARRIBA THEN
0006 salida:=salida-5
0007 END_IF
0008
0009 IF NOT (ESTA_ENMEDIO) AND ESTA_ABAJO THEN
0010 salida:=salida+5
0011 END_IF
```

Fig. B.11 Programa PLC_PRG.

B.2. Práctica 2

En esta segunda práctica, se insta al alumno a posicionar el objeto en las tres posiciones anteriormente descritas de forma continua. Comenzando por la posición "Arriba", desplazarlo a la posición "Medio" y posteriormente a la posición "Abajo" –Fig. B.12–.

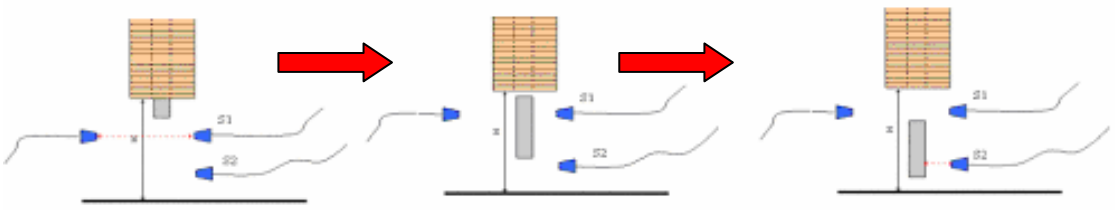

Fig. B.12 Posición del objeto émbolo para cada una de las posiciones

Para realizar esta segunda práctica, se utilizará lo aprendido en la práctica inicial.

En primer lugar se definen las tres posiciones –Fig. B.13–:

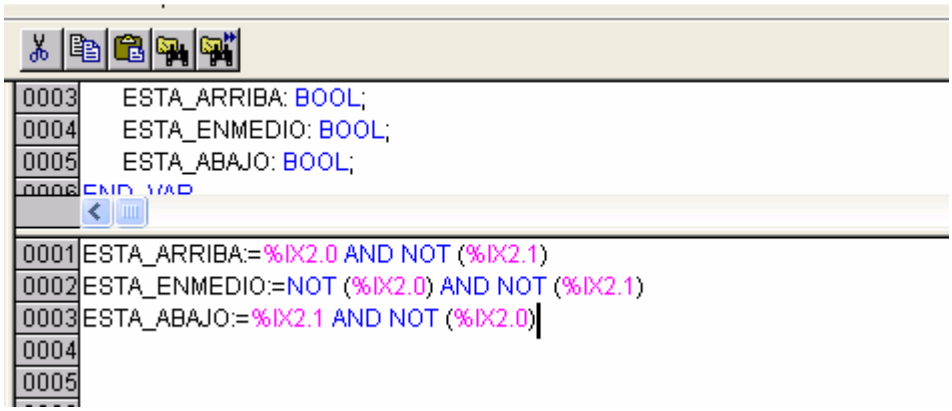

Fig. B.13 Programa PLC_PRG. Definición de estados

Seguidamente, se realiza la secuencia que se desea siga el objeto, se utiliza el lenguaje *GRAFCET* para la diseñar la misma. Se realiza una secuencia de 4 estados (el inicial y los tres posibles). La transición entre los estados vendrá fijada por un espacio de tiempo determinado –Fig. B.14–:

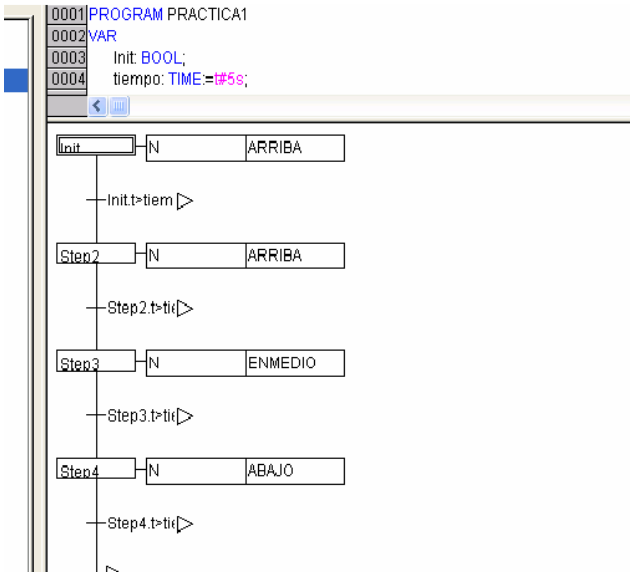

Fig. B.14 Programa practica1_PRG

Se puede observar como a una de las cuatro etapas se le asigna una acción, en este caso la

definición de un estado o posición.

Seguidamente se define como ha de actuar la salida analógica del autómata en cada caso -Fig. B.15-:

```
0005 PRATICA1;
0006
0007 IF ARRIBA AND NOT(ESTA_ARRIBA) THEN
0008     salida := salida + 10;
0009 END_IF;
0010 IF ABAJO AND NOT(ESTA_ABAJO) THEN
0011     salida := salida - 10;
0012 END_IF;
0013 IF ENMEDIO AND ESTA_ARRIBA THEN
0014     salida := salida - 10;
0015 END_IF;
0016 IF ENMEDIO AND ESTA_ABAJO THEN
0017     salida := salida + 10;
0018 END_IF;
0019
```

Fig. B.15 Programa PLC_PRG. Evolución de la salida

Notar que la salida irá variando en valor (aumentando o disminuyendo) en función de donde se encuentre el objeto y que posición le asigna la secuencia creada.

Para finalizar, se asignan unos límites (máximo y mínimo) a la variable salida –Fig. B.16–:

```
0021 IF sort>22000 THEN
0022 sort:=22000;
0023 END_IF;
0024 IF sort<5000 THEN
0025 sort:=5000;
0026 END_IF;
0027
```

Fig. B.16 Programa PLC_PRG. Límites de la salida

B.3. Practica 3

En la práctica anterior, el objeto se posicionaba en las tres posiciones posibles (Arriba, En medio y Abajo) para después volver a la posición inicial. En esta práctica se propone que el objeto vuelva a la posición inicial realizando el ciclo a la inversa –Fig. B.17–

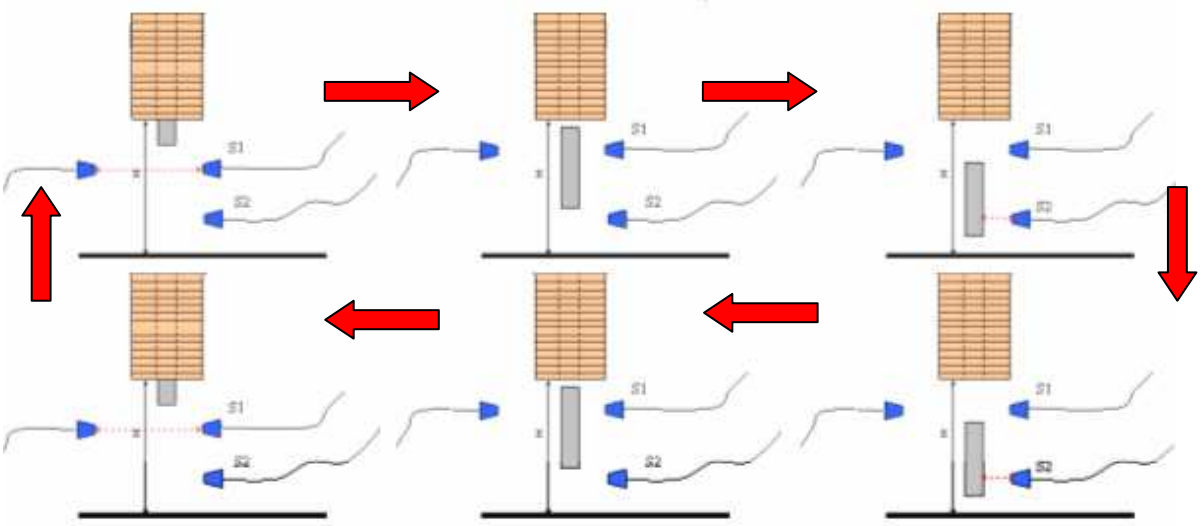

Fig. B.17 Posición del objeto émbolo para cada una de las posiciones

Cabe destacar que para la realización de esta práctica, solo es necesario cambiar la secuencia que seguirá el objeto ya que tanto la definición de los estados (posiciones) como el comportamiento de la salida del autómata serán idénticas. La nueva secuencia similar a la anterior, tan solo se le han de añadir 2 etapas –Fig. B.18–:

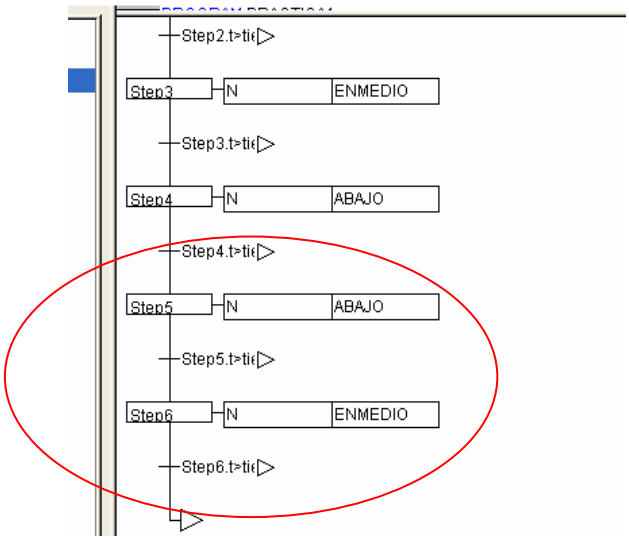

Fig. B.18 Programa practica3_PRG

B.4. Práctica 4

En esta cuarta práctica, se propone realizar un movimiento del objeto móvil (émbolo) sin tener en cuenta los sensores, es decir moverlo entre dos posiciones alternativamente (posición máxima superior y posición mínima inferior). De forma que el émbolo realice un movimiento como el siguiente -Fig. B.19-:

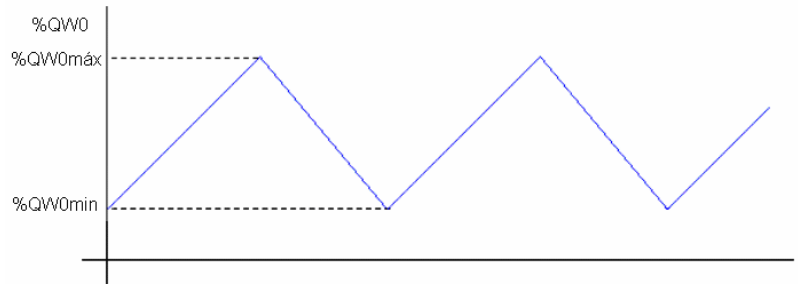

Fig. B.19 Evolución de la salida

Para realizar esta práctica, se procede como en los casos anteriores, en primer lugar se definen las variables que se utilizarán para realizar la secuencia -Fig. B.20-:

```
0001 PROGRAM PLC_PRG
0002 VAR
0003     salida AT %QW0: INT := 14000;
0004     ascendent: BOOL := TRUE;
0005     START: BOOL;
0006 END_VAR
```

Fig. B.20 Programa PLC_PRG. Definición de variables

Seguidamente se implementa el programa que definirá como ha de actuar la salida. En este caso no es necesario definir la secuencia a seguir con una aplicación *GRAFCET*, ya que la salida evoluciona de igual forma independientemente de la posición en la que se encuentre.

```
0001
0002 IF START THEN
0003 IF salida > 19000 THEN
0004 ascendent:=FALSE;
0005 END_IF
0006 IF salida < 6000 THEN
0007 ascendent:=TRUE;
0008 END_IF
0009
0010 IF salida < 20000 AND ascendent THEN
0011 salida:=salida+500;
0012 END_IF
0013
0014 IF salida > 5000 AND NOT (ascendent) THEN
0015 salida:=salida-500;
0016 END_IF
0017 END_IF
```

Fig. B.21 Programa PLC_PRG.

Seguidamente se realiza una aplicación visual que permita ver la evolución de la salida en un gráfico [2].

Para crear una nueva visualización hay que acceder a la pestaña *Visualizations* del árbol del proyecto y con el botón derecho hacer clic en la carpeta *Visualizations*→*Add object*… –Fig. B.22–

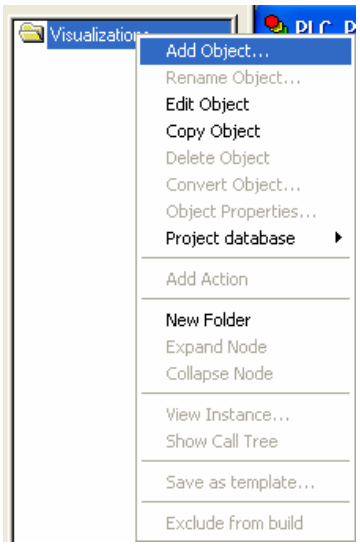

Fig. B.22 Insertar nuevo objeto de visualización

Seguidamente, aparecerá un menú para elegir el nombre de la visualización –Fig. B.23–.

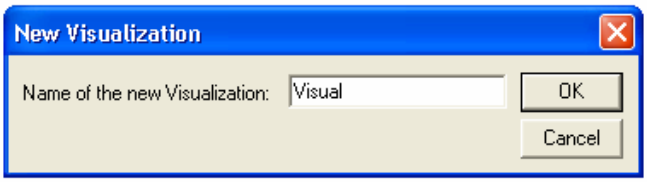

Fig. B.23 Escribir el nombre de la nueva visualización

A continuación se creará una ventana vacía –Fig. B.24– donde se podrán dibujar todos los elementos necesarios para poder simular el proyecto mediante la barra de elementos de visualización. Para insertar una gráfica se utiliza el botón de la barra de elementos. Una vez introducida se hace doble clic en ella y se configuran los ejes de coordenadas y las variables que se desean visualizar.

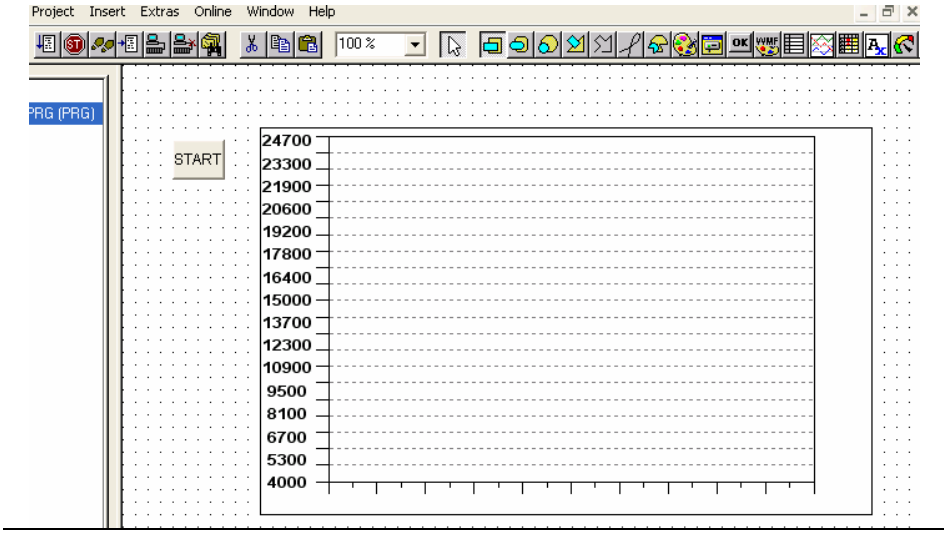

Fig. B.24. Gráfica creada en la visualización HMI

En cuanto al eje horizontal, al ser un sistema relativamente rápido se define un intervalo de 10 s con una separación de 1 s. En el eje de ordenadas correspondiente a la salida (%QW0) se medirá un rango definido entre 25000 y 4000.

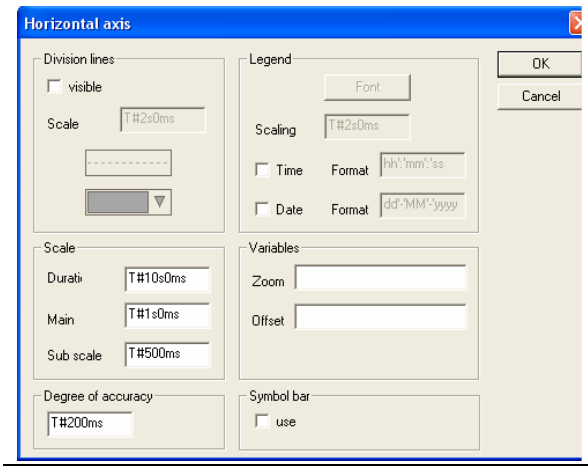

Fig. B.25. Configuración eje abscisas

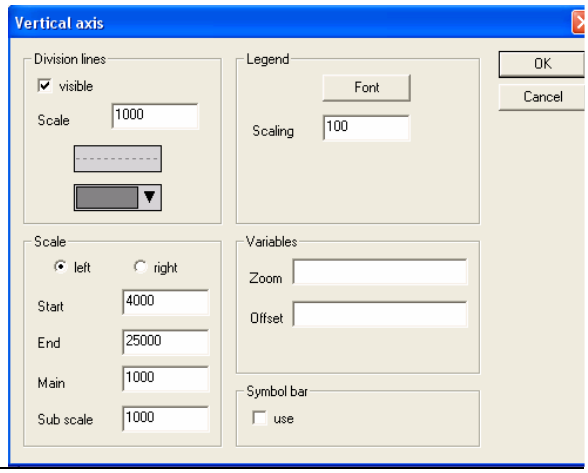

Fig. B.26.Configuración eje ordenadas

Destacar que se ha creado también un pulsador que servirá para activar y desactivar la secuencia. Mediante el elemento es posible dibujar botones que controlen diferentes variables de entrada –Fig. B.27–.

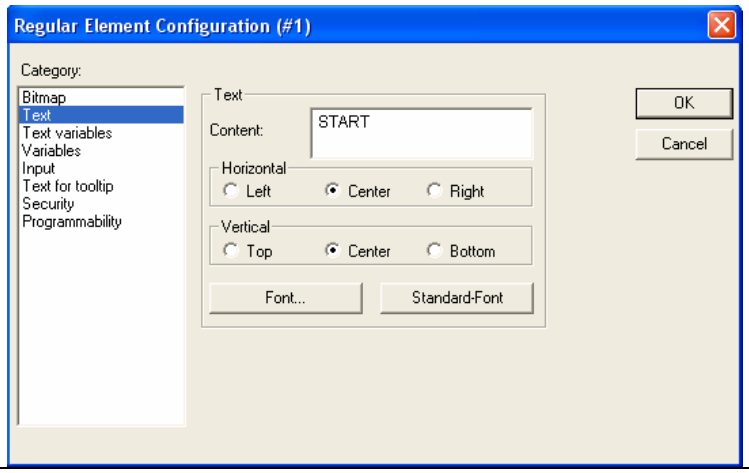

Fig. B.27. Gráfica creada en la visualización HMI

Para controlar el inicio de la secuencia se debe poder controlar el estado de la variable "*START*" que se activará/desactivará con cada clic del ratón. *Input* → *Toogle variable.* –Fig. B.28–

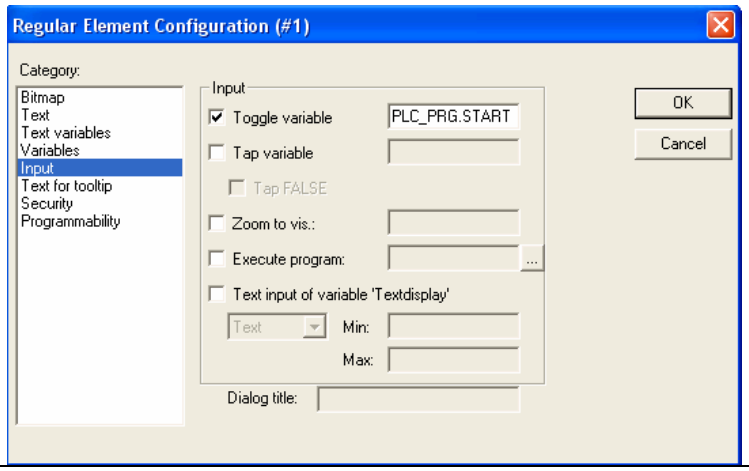

Fig. B.28. Opción del menú para crear el mando START.

Una vez configurada la visualización el resultado que se obtiene es el siguiente –Fig. B.29– :

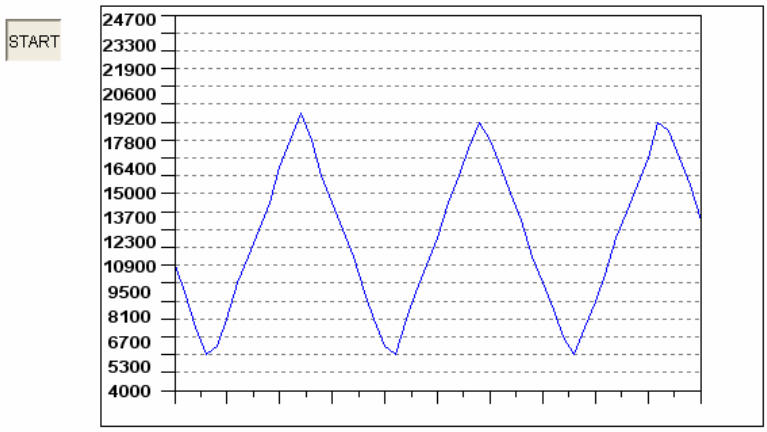

Fig. B.29 Evolución de la salida en la simulación

C. Valoración económica

El objetivo de este apartado es plantear una valoración económica lo más aproximada posible de los costes del proyecto.

La valoración económica se ha dividido en dos apartados, el coste del prototipo de la plataforma, y el coste de investigación y desarrollo, que incluye el desarrollo de la plataforma y el de la creación de las prácticas para el control de las estaciones didácticas.

C.1. Coste del prototipo

En este apartado se considerará el coste total de la construcción del prototipo de la plataforma incluyendo también el coste de la mano de obra. Por el contrario, no se tendrá en cuenta el coste de diseño y desarrollo, puesto que este se considera por separado y se comenta en el apartado C.2.

Si bien algunos de los elementos utilizados en la construcción de la plataforma han sido donados, como por ejemplo el PLC, y por tanto no han supuesto ningún coste, se ha considerado este coste con el objetivo de tener una idea lo más aproximada posible del coste.

Para calcular el coste de la mano de obra se supone un coste por hora de 25 €/h. El coste total del prototipo es de 1499,82 €.

En la siguiente tabla (Tabla C.1.1) se detallan los costes de construcción del prototipo:

Cantidad	Concepto	Coste unitario (€)	Coste (€)
1	Módulo Wago 750–431	52,84	52,84
1	Módulo Wago 750–466	180,18	180,18
1	Módulo Wago 750–530	59,61	59,61
1	Módulo Wago 750–550	191,28	191,28
1	Módulo Wago 750–600	12,82	12,82
1	Módulo Wago 750–841	440,26	440,26
2	Sensor de posición Osiris Xuda 1	45,06	90,12
1	Convertidor CC/CC	8,97	8,97
1	Fuente lineal (rectificador)	10,54	10,54
1	Fuente de alimentación FTE30V5A	100,00	250,00
1	Interruptor automático multi 9 15A	34,70	34,70
1	Soporte para el electroimán	5	5
-	Bobinado de cobre	8	8
-	Mano de obra (15 h)	375,5	375,5
	TOTAL	**1499,82**	

Tabla C.1.1. Coste del prototipo.

C.2. Coste de investigación y desarrollo

A los costes de construcción del prototipo se tienen que añadir los costes del tiempo dedicado a la investigación, al diseño y al desarrollo de las prácticas propuestas para ser realizadas con la plataforma construida.

También se deben tener en cuenta los costes asociados a los elementos necesarios para llevar a cabo estas tareas, como son el del *switch* o el de la cámara IP. Para calcular el coste de éstos se contempla una amortización contando una vida útil de 3 años y una duración del proyecto de 9 meses –25% del coste unitario–.

A este subtotal se le debe añadir el coste de investigación y desarrollo que ha representado el proyecto, aun cuando es muy difícil hacer una estimación de las horas dedicadas a la realización del proyecto, así como la división entre ellas. Se debe tener en cuenta que el precio hora del coste de personal es el precio de cobro de un ingeniero júnior. Estos costes quedan reflejados en la Tabla C.2.1:

Concepto	**Coste unitario (€)**	**Coste amortizado (€)**
Switch ETHERNET 8-P	94,14	23,54
Cámara IP	200	50
	Subtotal	73,54
Concepto	**Coste hora (€)**	**Coste total (€)**
Diseño (100 h)	48,00	4800,00
Investigación (200 h)	48,00	9600,00
Programación (50 h)	30,00	1500,00
Redacción (170 h)	20,00	3400,00
	Subtotal	19300,00
	TOTAL	**19373,54**

Tabla C.2.1. Coste de investigación y desarrollo.

C.3. Coste total del proyecto

El coste total del proyecto realizado, sumando los dos costes concretados en los puntos anteriores, se representa en la Tabla C.3.1.

El coste total de la realización del proyecto es de 20873,36€, del cual la mayor parte del coste es debido a los costes de investigación y desarrollo. La amortización de la investigación y el desarrollo del proyecto es del 100% en este prototipo, puesto que solo se ha fabricado uno. Si se fabricaran más unidades, estos costes se deberían repartir entre todas las unidades fabricadas.

Por otra parte, se debe tener en cuenta que este precio es el correspondiente por el desarrollo de un prototipo, y no el de un producto final comercializable en el mercado general.

Concepto	Coste (€)
Coste prototipo	1499,82
Coste de investigación y desarrollo	19373,54
TOTAL	**20873,36**

Tabla C.3.1. Coste total del proyecto.

D. Estudio del impacto ambiental

"La evaluación del impacto ambiental es un instrumento de carácter preventivo para la protección del medio ambiente. Permite conocer la incidencia de un proyecto sobre el entorno y adoptar medidas para evitar o corregir los impactos que podrían ocasionar su ejecución y funcionamiento.

Su aplicación, basada en el principio de que es mejor prevenir las perturbaciones en el origen que combatir después sus efectos, está recomendada por varias organizaciones internacionales, como las Naciones Unidas, la Organización Mundial de la Salud, la Organización para la Cooperación y el Desarrollo Económico (OCDE) y la Comunidad Económica Europea. La Comunidad Económica Europea aprobó el año 1985 la Directiva de Evaluación de Impacto Ambiental (85/377/CEE)."

Debido a la naturaleza del proyecto, no tiene demasiado sentido hacer un estudio de evaluación de impacto ambiental. La plataforma que ha resultado de este proyecto es un prototipo, no se ha pensado en hacer una comercialización del producto. Si llegado el caso, surgiera la posibilidad de comercializar la plataforma, se plantearía la idea de realizar dicho estudio.

En cambio, si se hace un pequeño estudio de impacto ambiental centrado en el impacto ambiental que puede causar este prototipo, se pueden tener en cuenta los siguientes puntos claves:

- Materiales utilizados
- Reciclaje

D.1. Materiales utilizados

La ley de impacto ambiental de ámbito europeo contempla una directiva que afecta a los materiales utilizados en aparatos electrónicos [5]. La directiva es la *2002/95/EC de Restricción de Uso de determinadas Sustancias Peligrosas (RUSP).*

Rohs (restricción del uso de determinadas sustancias peligrosas en aparatos electrónicos y eléctricos) [6] es una directiva de la Unión Europea que restringe el uso de 6 materiales peligrosos en la fabricación de varios tipos de equipos electrónicos y eléctricos. Esta directiva entró en vigor el día 1 de julio del 2006. Se prohíbe superar unos límites de

concentración máximos de dichas sustancias. Estas substancias son: plomo (Pb), Mercurio (Hg), Cadmio (Cd), Cromo hexavalente (Cr(VI)) y un tipo de materiales ignífugos que se usan en ciertos plásticos como el PBB y el PBDE.

Todos los componentes electrónicos que se han utilizado para implementar las placas de este proyecto (convertidores CC/CC, fuente lineal) cumplen la normativa *Rohs*. Además, el tratamiento final de los materiales utilizados durante la creación de las PCB (líquido revelador, ácido clorhídrico y perborato sódico) cumple la directiva *91/689/CEE, de 12 de diciembre de 1991, relativa a la gestión de residuos peligrosos*.

D.2. Reciclaje

Existe otra directiva europea que se encarga de la recuperación de equipos electrónicos y eléctricos, es la *2002/96/EC de Residuos de Aparatos Electrónicos y Eléctricos (RAEE)* 0. Esta directiva tiene como objetivo reducir la cantidad de estos residuos y la peligrosidad de los componentes, fomentar la reutilización de los aparatos y la valorización de sus residuos y determinar una gestión adecuada intentando mejorar la eficacia de la protección ambiental.

Asimismo, se pretende mejorar el comportamiento medioambiental de todos los agentes que intervienen en el ciclo de vida de los aparatos eléctricos y electrónicos, por ejemplo, los productores, distribuidores y consumidores, y, en particular, de aquellos agentes directamente implicados en el tratamiento de los residuos derivados de estos aparatos.

Para conseguir estos objetivos se establecen una serie de normas aplicables a la fabricación del producto y otras relativas a una correcta gestión ambiental cuando se convierta en residuo. Esta directiva entró en vigor el 13 de agosto del 2005.

BIBLIOGRAFÍA

Referencias bibliográficas

[1] LORENZO, N. *Aplicación docente de una plataforma de accionamientos mecatrónicos controlada a través de Internet*. PFC Barcelona, ETSEIB 2007

[2] CITCEA. Formación. [http://formacio.citcea.upc.edu/, 2 de Febrero 2007]

[3] SENSORES OPTO-ELECTRONICOS. La luz como medio detector [http://www.nortecnica.com.ar/pdf/teoria_opticos_2_2.pdf, 2 de Febrero 2007]

[4] AUTÓMATAS.ORG [http://www.automatas.org/redes/grafcet.htm, 2 de Febrero 2007]

[5] INNOVACIÓN EUROPEA .[http://www.cdc.fonacit.gov.ve/DB/ conicit/ edocs/2006/fulltext/itt05-3_es.pdf, 6 de Marzo 2007]

[6] WIKIPEDIA. Rhos. [http://es.wikipedia.org/wiki/Rohs, 28 de Marzo de 2007]

[7] RED DE TECNOLOGÍAS Y PRODUCOTS SANITARIOS. [http://www.tecnologias- sanitarias.com/DIR_WEEE/index.htm, 28 de Marzo de 2007]